SYSTEM AND METHOD FOR PARSING A DOCUMENT

Background of the Invention

This invention relates generally to a system and method for processing a document and in particular to a system and method for identifying a plurality of phrases within the document which indicate the context of the document.

Various factors have contributed to the extensive storage and retrieval of textual data information using computer databases. A dramatic increase in the storage capacity of hard drives coupled with a decrease in the cost of computer hard drives, and increases in the transmission speed of computer communications have been factors. In addition, the increased processing speed of computers and the expansion of computer communications networks, such as a bulletin board or the Internet, have been factors. People therefore have access to the large amounts of textual data stored in these databases. However, although the technology facilitates the storage of and the access to the large amounts of textual data, there are new problems that have been created by the large amount of textual data that is now available.

In particular, a person trying to access textual data in a computer database having a large amount of data needs a system for analyzing the data in order to retrieve the desired information quickly and efficiently without retrieving extraneous information. In addition, the user of the system needs an efficient system for condensing each large document into a plurality of phrases (one or more words) which characterize the document so that the user of the system can understand the document without actually viewing the entire document. A system for

condensing each document into a plurality of key phrases is known as a parsing system or a parser.

In one typical parser, the parser attempts to identify phrases which are repeated often within the document and identifies those phrases as being key phrases which characterize the document. The problem with such a system is that it is very slow since it must count the repetitions of each phrase in the document. It also requires a large amount of memory. As the amount of data to be parsed increases, the slow speed of this parser becomes unacceptable. Another typical parser performs a three step process to identify the key phrases. First, each word in the document is assigned a tag based on the part of speech of the word (i.e., noun, adjective, adverb, verb, etc.) and certain parts of speech, such as an article or an adjective, may be removed from the list of phrases which characterizes the document. Next, one or more sequences of words (templates) may be used to identify and remove phrases which do not add any understanding to the document. Finally, any phrase which is an appropriate part of speech and does not fall within one of the templates is accepted as a key phrase which characterizes the document. This conventional parser, however, is also slow which is unacceptable as the amount of data to be parsed increases.

In all of these conventional parser systems, the parser attempts to break the document down into smaller pieces based on the characteristics (frequency of repetition or part of speech) of the particular words in the document. The problem is that language generally is not that easily classified and therefore the conventional parser does not accurately parse the document or requires a large amount of time to parse the document. In addition, the conventional parser

systems are very slow because they all attempt to use complex characteristics of the language as a method for parsing the key phrases out of the document. These problems with conventional parsers becomes more severe as the number of documents which must be parsed increases. Today, the number of documents which must be parsed is steadily increasing at a tremendous rate due to, among other things, the Internet and the World Wide Web. Therefore, these conventional parsers are not acceptable. Thus, it is desirable to provide a parsing system and method which solves the above problems and limitations with conventional parsing systems and it is to this end that the present invention is directed.

Summary of the Invention

A parser system and method in accordance with the invention is provided in which the break characters within a sentence or a paragraph are used to parse the document into a plurality of key phrases. The parser system in accordance with the invention is very fast and does not sacrifice much accuracy for the speed. The break characters within the document may include punctuation marks, certain stop words and certain types of words such as verbs and articles. The parser system may include a buffer which receives one or more words before it receives a break character. When the buffer receives a break character, the parser may determine whether the phrase before the break character is saved based on the type of break character. In particular, if the break character is a punctuation mark, the parser may keep the one or more words before the break character as a key phrase. If the break character is another type of character, the phrase before the break character may or may not be saved. Once the fate of the phrase has been determined, the buffer is flushed and the next sequence of one or more words is read into the

buffer so that it may also be parsed. In this manner, a plurality of phrases in the document may be rapidly extracted from the document based on the break characters within the sentences and paragraphs of the document.

The parser system in accordance with the invention may also be used to parse various different foreign languages into phrases provided that the rules database includes rules that are applicable to the particular foreign language. In particular, each foreign language may have slightly different syntax or characters (in the case of Asian languages or Arabic, for example) so that the rules must reflect those syntactic and character differences.

Thus, in accordance with the invention, a system for parsing a piece of text into one or more phrases which characterize the document is provided. The system comprises a buffer for reading one or more words from the piece of text into the buffer and a parser for identifying a phrase contained in the buffer, the phrase being a sequence of two or more words in between break characters. The parser further comprises means for determining the type of break character that follows the identified phrase and means for saving a key phrase from the buffer based on the determined type of break character. The key phrases are stored in a database.

In accordance with another aspect of the invention, the parsing method may include a two-pass process wherein phrases are extracted from the piece of text as described above. During the second pass, all of the occurrences of the extracted phrases in the piece of text are retrieved. The second pass ensures that phrases that were not extracted at each location in the piece of text may still be retrieved.

Brief Description of the Drawings

Figure 1 is a block diagram of a text processing system;

Figure 2 is a block diagram of a parsing system in accordance with the invention;

Figure 3A is a flowchart illustrating a two-pass parsing method in accordance with the invention;

Figure 3B is a flowchart illustrating more details of the extracting phrases step of the parsing method shown in Figure 3A;

Figure 4 is an example of a document to be parsed by the parsing system in accordance with the invention;

Figures 5A - 5L are diagrams illustrating the operation of the parsing buffer in accordance with the invention on the document shown in Figure 4;

Figure 6 is a diagram illustrating a piece of Japanese text; and

Figure 7 is a diagram illustrating the Japanese phrases extracted from the Japanese text of Figure 6 in accordance with the invention.

Detailed Description of a Preferred Embodiment

The invention is particularly applicable to a system for parsing English language documents and it is in this context that the invention will be described. It will be appreciated, however, that the system and method in accordance with the invention has greater utility, such as

to other languages and to various different pieces of textual data. To better understand the invention, a text processing system will now be described.

Figure 1 is a block diagram of a text processing system 10. The text processing system 10 may include a parser system 12, a clusterizer 14, a map generator 16 and a database (DB) 18. The text processing system may receive one or more pieces of text , such as stories, press releases or documents, and may generate a map graphically showing the relationships between the key phrases in the document. Each piece of text may be received by the parser system 12 which processes each piece of incoming text and generates one or more key phrases for each piece of text which characterizes the piece of text. The key phrases may be stored in the database 18. The details about the parser system will be described below with reference to Figures 2- 5. Once the key phrases are extracted from each piece of text, the clusterizer 14 may generate one or more clusters of the key phrases based on the relationships between the phrases. The clusters generated may also be stored in the database 18. The map generator 16 may use the generated clusters for the pieces of text in the database in order to generate a graphical map showing the relationships of the key phrases within the various pieces of text in the database to each other so that a user of the system may easily search through the database by viewing the key phrases of the pieces of text. More details about the clusterizer and map generator are disclosed in co-pending U.S. patent application serial no. 08/801,970 which is owned by the assignee of the present invention and is incorporated herein by reference. The text processing system may be implemented in a variety of manners including a client/server type computer system in which the client computers access the server via a public computer network, such as the Internet. The

parser, the clusterizer and the map generator may be software applications being executed by a central processing unit (not shown) of the text processing system 10. Now, the parser system 12 in accordance with the invention will be described in more detail.

Figure 2 is a block diagram of the parsing system 12 in accordance with the invention. The parsing system 12 may include a buffer 20, a parser 22 and a rules database (rules DB) 24. The buffer may store one or more words of the incoming piece of text, which may be a document, which are analyzed by the parser 22 using the rules contained in the rules DB 24. The output of the parser system 12 is one or more phrases (each phrase containing one or more words) which characterize the document being parsed. In particular, the parser may separate phrases in the document based on break characters within the document in accordance with the invention. In more detail, one or more words may be read into the buffer from the document until a break character is identified. Thus, the parser system 12 identifies phrases which are between break characters. Then, based on the type of break character, the phrase may be saved as a key phrase or deleted. The parser system 12, for example, may be implemented as one or more pieces of software being executed by a microprocessor (not shown) of a server computer which may be accessed by a plurality of client computers over a computer network, such as the Internet, a local area network or a wide area network. The parser 22 advantageously rapidly extracts key phrases from a piece of text using break characters. The break characters in accordance with the invention will now be described.

The break characters may include an explicit break, such as a punctuation mark, numbers, words containing numbers, and stop words. The stop words may be further classified as soft stop

words or a hard stop words. Each of these different break characters will now be described. The explicit break characters may include various punctuation symbols, such as a period, a comma, a semicolon, a colon, an exclamation point, right or left parenthesis, left or right square brackets, left or right curly braces, a return character or a line feed character. The stop characters may be a generated list or it may include a slash (/) and an ampersand symbol (@). A separator may be defined as digits, letters, foreign characters, break characters, apostrophes, dashes and other stop characters. The various words in a piece of text may be categorized as articles, connectors, hard and soft stop characters, linguistic indicators, a syntactic categories such as nouns, verbs, irregular verbs, adjectives and adverbs.

In parsing the characters in the piece of text, separators may always be added to a phrase. A apostrophe or dash at the beginning of a word is treated as a break character (see below), an apostrophe or dash at the end of a word is also treated as a break character and a word with an apostrophe or dash in the middle of the word is added to the phrase in the buffer. All stop characters and breaks are treated as stop characters and breaks as described below. At the word level of parsing, proper nouns are retained by testing for an upper case letter at the first character of the word. In addition, all words with only upper case letters and numeric words are kept in the buffer. Optionally, a numeric string may be classified and treated as a stop character. The following are mandatory word level parsing rules. First, the word following as possessive "s" may be deleted. For example, as the sentence "The cat's paw is wet." is parsed in accordance with the invention, "the" is deleted and "cat" is put into the buffer and then deleted when the break character (the aprostrophe) is detected. The apostrophe is deleted because it is punctuation

and then the next character to parse is the possessive "s" after the apostrophe which is deleted along with the word "paw" since it follows the possessive "s". Connector words appearing at the beginning of a phrase are also deleted although a connector word followed by "The" is kept in the buffer. For a hard stop character, the last phrase connected to the hard stop character is deleted and the remaining buffer is processed. A soft stop character may be treated as a break character. A repeated character is treated as a stop character.

To further remove unwanted words for parsing. some optional phrase level parsing rules may be used. In particular, phrases longer than a predetermined length, such as six words, may be deleted, a phrase with all upper case words may be deleted and a phrase with all numeric words may be deleted. All of the above parsing rules may be stored in the parsing rules database 24 shown in Figure 2. Now, the details of the parser system 12 will now be described with reference to Figures 3A and 3B.

Figure 3A is a flowchart illustrating a two pass parsing process 30 in accordance with the invention. In particular, during a first pass 40, one or more phrases are extracted from a piece of text using the hard and soft stop words as described below with reference to Figure 3B. The first pass thus extracts noun phrases. For example, if a piece of text includes, "The big frog and the kangaroo jumps down.", the first pass extracts the phrase "big frog", but not "kangaroo jumps" as described below. During a second pass 41, all extracted phrases are retrieved from the piece of text. In particular, the occurrence of each extracted phrase in the piece of text may be retrieved from the piece of text. For example, assume that a piece of text contains the fragments, "The software bugs on..." and "software bugs are...". The parser in the first step throws away

the first occurrence of the term "software bugs" since it is followed by a hard stop, but retains the second occurrence since it is followed by a soft stop. To prevent the parser from discarding good noun phrases, such as the first occurrence of the term "software bugs", the second pass retrieves all occurrences of the extracted phrases from the piece of text so that, for example, both occurrences of the term "software bugs" are retrieved. Now, the first pass of the method will be described in more detail.

Figure 3B is a flowchart illustrating more details of the phrase extracting step 40 for parsing a document in accordance with the invention. The method begins as a first word of the document is loaded into the buffer from a document database or a memory of the server in step 42. Next, the parser determines if the word is a break character in step 44. The parser may also delete certain characters or words at this stage of the parsing process. If the word is not a break character, the method loops back to step 42 and the next word of the document is read into the buffer. This process of reading a word into the buffer is repeated until a break character is encountered so that the buffer contains a sequence of words (a phrase) which has a break character before the sequence of words and a break character after the sequence of words. In this manner, the document is parsed into phrases which are separated from one another by break characters.

If a break character is encountered, the parser may determine if the break character is an explicit break character in step 46, delete the break character and extract the phrase contained in the buffer if an explicit break character exists in step 48. The phrase extracted from the buffer may be stored in a database for future use. Next, in step 50, the buffer may be flushed to empty

the words from the buffer and the buffer may begin loading new words into the buffer in steps 42 and 44 until another break character is identified. Returning to step 46, if the break character is not an explicit break character, the parser determines if the break character is a soft stop word in step 52. If the break character is a soft stop word, then the soft stop word is deleted and the phrase in the buffer is stored in the database in step 54, the buffer is flushed in step 50 and the buffer is refilled with new words from the document. If the break character is not a soft stop word (i.e., the break character is a hard stop word), the hard stop word and the phrase in the buffer are deleted in step 56, the buffer is flushed in step 50 and refilled with new words from the document in steps 42 and 44. In this manner, phrases from the document are extracted in accordance with the invention using the break characters and the type of break character to separate the phrases from each other and determine which phrases are going to be saved in the database. The parser in accordance with the invention does not attempt to analyze each word of the document to identify key phrases as with conventional systems, but does extract phrases from the document more quickly than conventional parsers and with as much accuracy as the conventional parsers. Now, an example of the operation of the parser in accordance with the invention will be described with reference to Figures 4 and 5A - 5L.

Figure 4 is an example of a document 60 to be parsed by the parsing system in accordance with the invention while Figures 5A - 5L illustrate the operation of the buffer during the parsing of the document 60 shown in Figure 4. In this example, the document is a short electronic news story, but the parser may also extract phrases from any other piece of text. In fact, the parser in accordance with the invention may be able to extract phrases from various

types of documents at speeds of up to 1 MByte of data per second. The particular story shown describes a new "snake-like" robot developed by NEC. Figures 5A - 5L illustrate, in a table 68, the operation of the buffer in accordance with the invention on the above story. In particular, a first column 70 of the table contains the current word being read into the buffer, a second column 72 contains the determination of the type of word by the parser in accordance with the invention, a third column 74 contains the contents of the buffer at the particular time, a fourth column 76 contains the word index (i.e., the phrases which are being extracted from the document) and a fifth column 78 contains comments about the parsing process.

As shown in Figure 5A, the first word read into the buffer is a sequence of asterisks at the beginning of the story which are classified by the parser as a break word (punctuation) and deleted from the buffer. The next word is "Computer" which is entered into the buffer since it is not a break word and the next word, which is "Select" is also entered into the buffer since it is also not a break word. Thus, the buffer contains the phrase "Computer Select" as shown in a cell 80. The next word in the document is a comma which is classified as a break character by the parser. Because the break character is punctuation (an explicit break), the words in the buffer are saved in the database as shown in the Word Index column 76 and the buffer is flushed. Now, new words are read into the buffer and parsed. The next word into the buffer is "October" which is a hard stop word because it relates to a date and it is deleted. The next word received by the buffer is "1995" which is a break character since it is a number and it is also deleted. The next word received by the buffer is "COPYRIGHT" which is identified as a stop word because it is all capital letters and it is deleted. The next word is "Newsbytes" which is not a break character and

is therefore stored into the buffer. The next word is "Inc." which is also stored in the buffer. The next word is a period which is a break character so that the buffer contents "Newsbytes Inc." are saved into a database as shown in the Word Index column, the break character is deleted and the buffer is flushed.

The next two word received by the buffer, which are "1995" and a sequence of asterisks, are both break words which are deleted. The next two words received by the buffer are "Newsbytes" and "Newsbytes" which are both stored in the buffer. The next word received is "August" which is a hard stop word so that the contents of the buffer and the hard stop word is deleted. The next three words received by the buffer are all break characters (i.e., numbers or punctuation) which are deleted. The next word is a word containing a number in a cell 82 which is stored in the buffer, but then deleted when the next character is a break character because the buffer only contains a single word. As can be seen in Figures 5B - 5L, the parsing process continues for the entire document so that a list of key phrases, as shown in the Word Index column 76, are extracted from the document and saved in a database.

In summary, phrases which characterize the document or piece of text may be rapidly extracted from the document in accordance with the invention. The invention uses the break characters in the document or the piece of text to separate the phrases from each other and to extract the key phrases for a document. In the example above, the extracted phrases, such as "Newsbytes Inc.", "snake-like robot", "NEC Corporation", "robotically controlled electronic snake", "disaster relief work" and "world's first active universal joint" permit a person reviewing only the key phrases to understand the context of the document without reviewing the entire

document. The parsing system in accordance with the invention performs the extraction of the key phrases more rapidly than any other conventional parsing systems which is important as the total amount of textual data and documents available for parsing increases at an exponential rate due, in part, to the explosion of the user of the Internet.

The parser in accordance with the invention may be used to parse documents in various different foreign languages with minor modifications to the rules database to reflect changes in the characters and changes in the syntax of the language. To better understand this, an example of a piece of Japanese text is described along with the resulting Japanese taxonomy. However, the invention may be used with a variety of different foreign languages with minor modifications to the rules database.

Figure 6 is an example of a piece of Japanese text 90 while Figure 7 illustrates a list 92 of phrases 94 that have been extracted from the piece of Japanese text using the two-pass parsing method in accordance with the invention.

While the foregoing has been with reference to a particular embodiment of the invention, it will be appreciated by those skilled in the art that changes in this embodiment may be made without departing from the principles and spirit of the invention, the scope of which is defined by the appended claims.